Volume xx, (200y), Number z pp. 000–000

Multiresolution Surfaces Having Arbitrary Topologies by a Reverse Doo Subdivision Method

Faramarz F. Samavati[†]

Department of Mathematical Sciences
Shahid Beheshti University
Tehran, Iran
and
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada

Nezam Mahdavi-Amiri[‡]

Department of Mathematical Sciences
Sharif University of Technology
Tehran, Iran

Richard H. Bartels[§]

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

Abstract

We have shown how to construct multiresolution structures for reversing subdivision rules using global least squares models [16]. *As a result, semiorthogonal wavelet systems have also been generated. To construct a multiresolution surface of an arbitrary topology, however, biorthogonal wavelets are needed. In* [1] *we introduced local least squares models for reversing subdivision rules to construct multiresolution curves and tensor product surfaces, noticing that the resulting wavelets were biorthogonal (under an induced inner product). Here, we construct multiresolution surfaces of arbitrary topologies by locally reversing the Doo subdivision scheme. In a Doo subdivision, a coarse surface is converted into a fine one by the contraction of coarse faces and the addition of new adjoining faces. We propose a novel reversing process to convert a fine surface into a coarse one plus an error. The conversion has the property that the subdivision of the resulting coarse surface is locally closest to the original fine surface, in the least squares sense, for two important face geometries. In this process, we first find those faces of the fine surface which might have been produced by the contraction of a coarse face in a Doo subdivision scheme. Then, we expand these faces. Since the expanded faces are not necessarily joined properly, several candidates are usually at hand for a single vertex of the coarse surface. To identify the set of candidates corresponding to a vertex, we construct a graph in such a way that any set of candidates corresponds to a connected component. The connected components can easily be identified by a depth first search traversal of the graph. Finally, vertices of the coarse surface are set to be the average of their corresponding candidates, and this is shown to be equivalent to local least squares approximation for regular arrangements of triangular and quadrilateral faces.*

Keywords: *Multiresolution; Doo Subdivision; Surfaces; B-splines; Data Fitting.*

1. Introduction

The objective of a multiresolution technique is to obtain representations of curves, images and surfaces as a hierarchy of successively finer approximations in such a way that one can easily change one approximation to another. To achieve this, it is necessary to have an appropriate reverse procedure, transforming a given fine (or high) resolution model to a coarse (or low) one. To do this efficiently, we need a convenient way to store information about approximation errors compactly. This error information is usually accounted for by a linear combination of functions named *wavelets*. The process of transforming the approximate high resolution of the model to its corresponding approximate low resolu-

[†] Samavati's work was in part supported by a grant from IPM.
[‡] Mahdavi-Amiri's work was in part supported by a grant sponsored by the Research Council of Sharif University of Technology.
[§] Bartels' work was in part supported by NATO and by the National Science and Engineering Research Council of Canada.

tion, along with the determination of the coefficients of the wavelets is termed *decomposition*. The reconstruction process of obtaining the fine (high resolution) model from its corresponding coarse (low resolution) one and its error information is named *reconstruction*.

Finding multiresolutions of surfaces with arbitrary topologies is a well known problem having various applications in computer graphics [14]. Here, we intend to present a novel method for solving this problem by extending the ideas of reverse subdivision schemes introduced in [1], and reversing the Doo subdivision rule, which is a common subdivision procedure for surfaces of arbitrary topological type.

Subdivision methods are quite useful for generating curves and surfaces in computer graphics. There are several efficient varieties of subdivision that can be used for complex surfaces of arbitrary topology [3, 6, 7, 11, 19]. A fine approximation of the model is obtained by successively subdividing the vertices of a coarse approximation. This process is achieved by linear operations; each subdivision could be interpreted as a matrix transformation applied to the vertices. That is, if V^k are the vertices at the kth stage, and V^{k+1} is obtained by a subdivision of V^k, then

$$V^{k+1} = P^k V^k, \tag{1}$$

where V^k is an r-vector, V^{k+1} is an s-vector ($s > r$), and P^k, $s \times r$, is the *subdivision matrix*. P^k depends on k. In subdivision schemes each point of V^{k+1} depends upon only a few points of V^k that are situated in geometric proximity, and this implies that P^k is sparse. Under the arrangement that the points of the vectors are ordered according to geometric proximity (trivially done for curves but an open problem for arbitrary surfaces), P^k will be a banded matrix. For the simple subdivision of curves, the banded form of P^k is one in which each column is a shifted version of the previous one (except for a few initial and final columns); hence, it is easy to generate.

In multiresolution, V^{k+1} is usually at hand as a high resolution approximation for a model, and it becomes necessary to decompose V^{k+1} to V^k, a low resolution approximation, and E^k, the coefficients of the wavelets. In general, it is not mandatory that V^{k+1} have been obtained as a result of a subdivision. For an efficient handling of the models, the following properties are desired [18, 14, 17]

- V^k is a good approximation for V^{k+1}.
- The storage requirement for storing V^k and E^k is not more than that of V^{k+1}.
- The time required to decompose V^{k+1} into V^k and E^k is linearly dependent on s.
- The time required to reconstruct V^{k+1} from V^k and E^k is linearly dependent on s.

The processes of decomposition and reconstruction are all linear operations; they can be represented by the following relations:

$$\begin{bmatrix} A^k \\ B^k \end{bmatrix} V^{k+1} = \begin{bmatrix} V^k \\ E^k \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} P^k & Q^k \end{bmatrix} \begin{bmatrix} V^k \\ E^k \end{bmatrix} = V^{k+1}, \tag{3}$$

where

$$\begin{bmatrix} A^k \\ B^k \end{bmatrix} \begin{bmatrix} P^k & Q^k \end{bmatrix} = \begin{bmatrix} I & O \\ O & I \end{bmatrix}. \tag{4}$$

Therefore, if there are convenient representations for Q^k, P^k, B^k and A^k, then the operations of decomposition (2) and reconstruction (3) can easily be carried out. In practice, such specifications as bandedness (with a small band width) of the matrices and simplicity of the components are not always obtainable. Such matrices presented in [16] are obtained by a reverse subdivision scheme that makes use of least squares fitting. The approach in [16] does not make use of A^k and B^k directly, since these matrices are usually be expected to be full for least squares. Furthermore, in solving the least squares problem for multiresolution of surfaces having arbitrary topologies (non tensor product forms), we encounter matrices with many nonzeros (not necessarily banded), and thus the work of decomposition and reconstruction may not be linear in the data.

Equations (2) and (3) appear in the literature on wavelets. P^k embodies the representation of basis functions (*scale functions*) for a space $\mathcal{V}^k$ in terms of basis functions for a containing space $\mathcal{V}^{k+1}$. Q^k gives a corresponding representation for basis functions (*wavelets*) for $\mathcal{V}^{k+1} - \mathcal{V}^k$ in terms of the basis for $\mathcal{V}^{k+1}$. A^k and B^k provide corresponding expressions for the dual bases of the dual spaces $\tilde{\mathcal{V}}^k$ and $\tilde{\mathcal{V}}^{k+1}$. Equation (4) defines the primal/dual, scale/wavelet system as *biorthogonal*. The V^k and V^{k+1} represent coefficients of the best approximations of some function f^∞ in terms of functions f^k in $\mathcal{V}^k$ and f^{k+1} in $\mathcal{V}^{k+1}$, respectively. In the geometric setting, however, the V^k and V^{k+1} are themselves of primary interest as points in n-dimensional Euclidian space. We neither know nor care about the spaces $\mathcal{V}^k$ or $\mathcal{V}^{k+1}$, about bases on them, functions in them, or best-approximation norms over them. Our attention is fixed on the Euclidian spaces of the points C and the least-squares norm in the simple geometry of those spaces.

In [1], we have proposed a local reverse subdivision process to obtain banded A^k and B^k for tensor product curves and surfaces. Most importantly, the bandedness of all the matrices A^k, B^k, P^k, and Q^k is equivalent to the *local* character of the decomposition and reconstruction processes. Every point of V^k depends on only a combination of some small number of nearby points of V^{k+1}. Every wavelet coefficient E^k is likewise such a combination, and every point of V^{k+1} is a com-

bination of nearby points in V^k and associated coefficients E^k. The decomposition can be carried out by appealing to these local combinations, and the underlying matrices can be ignored.

In this paper we describe a process to handle surfaces of arbitrary topologies. Our process will use P^k implicitly in the form of the local operations of the Doo subdivision scheme. The elements of V^k will be provided by a local average process that implicitly defines A^k. B^k will also be implicit in operations that compute the E^k from differences between elements of V^{k+1} and V^k. Finally, Q^k will have such a simple regular form that it can also be handled as local operations on the data. Thus, none of the matrices in equations (2) through (4) will have to be dealt with explicitly.

2. Related Works

In this paper, we construct multiresolution surfaces of arbitrary topology. We do this work on surfaces that have the connectivity that may result from Doo subdivision, although we make no demand that the vertices of the surface mesh actually come from such a subdivision. They can have been measured from an object using some 3D picking device, for example.

The general viewpoint of our work is close to the viewpoint of Lounsbery et al. [14]. We suppose there is a sequence of meshes $M^0, M^1, \ldots, M^k, \ldots$ and that the elements of each provide different approximations of a limit surface S. This sequence of meshes is to be created by a subdivision rule, possibly with some error at each stage in each vertex that we are prepared to record separately. The topology of S, and all M^k for $k \geq 1$, is inherited from M^0; and therefore, all M^k and S have the same topology. But M^0 can have any topology, such as an arbitrary genus, and any kind of continuity and connectivity. This generality is considerable, but there is a restriction: the mesh connections and adjacencies in M^{j+1} must be obtainable from subdividing M^j. This condition is called *subdivision connectivity*, and it restricts the mesh adjacency structure; however, it doesn't restrict the position of the vertices in mesh.

A *multiresolution surface* associated with this sequence of meshes, as we wish to construct it, will start with some mesh M^{k+1} that approximates a surface S. We assume that M^{k+1} has subdivision connectivity consistent with Doo subdivision, but we do not assume that M^{k+1} was produced by Doo subdivision or that S is the limit we would obtain if we apply Doo subdivision to M^{k+1} infinitely often. We wish to find a surface M^k such that applying Doo subdivision to M^k will produce M^{k+1} within an error E^{k+1}: $Doo : M^k \rightarrow M^{k+1} + E^{k+1}$. We wish to construct M^k so that E^{k+1} is small, and so that the storage required for M^k and E^{k+1} together is essentially the same as that required by M^{k+1}. The multiresolution surface to be constructed consists of $\{M^0, E^1, E^2, \ldots, E^{k+1}\}$ formed by repeating this process.

In [14], multiresolution of a surface of general topology is obtained by presenting a new class of subdivision wavelets. By contrast, we develop multiresolution of such surfaces by reversing subdivision as was discussed in [16] and [1] for curves and tensor-product surfaces. This contrast is similar to that between traditional methods of refining B-spline and Bezier models and the subdivision methods based upon B-spline and Bezier recursion. In subdivsion methods, linear and local refinement operations are carried out on the control vertices of a mesh instead of applying recursions to sums of basis functions. Although the basis functions (*scale functions* and *wavelets*) don't appear directly in the subdivision process, they have influence on the subdivision structure. The same view can be taken of our method with respect to the method of [14]. We construct the decomposition of a model by using linear and local operations without the direct use of scale and wavelet functions, although they do exist in the background of our construction.

A distinction should be made here between *multiresolution meshes* and *progressive meshes*, although the distinction is one of degree and not of kind. Most progressive mesh methods are based on edge, half edge and vertex pair collapsing [8, 10, 12], and since they simplify a mesh one edge at a time, there is a minimal difference between two successive meshes M^j and M^{j+1}, often only a diference in a couple of vertices. In contrast, the difference between two successive meshes in a multiresolution will usually involve all, or a large number, of the faces and vertices. It is not unusual that the number of vertices and faces in M^j will be only a fraction of the number in M^{j+1}. Progressive meshes work locally, while multiresolution techniques work globally, or at least over large regions of a mesh at once. In this regard, progressive mesh techniques can be used more readily at locations where a mesh has unusual features [10], while multiresolution and subdivision models have been explored more often to achieve multilevel editing [10, 19] as well as the compression of geometry and associated surface attributes such as color [4, 10]. Progressive models do not have the requirement of subdivision connectivity, but this requirement has not proved to be a restrictive condition in modeling.

3. Chaikin's Subdivision

Since Doo subdivision may be explained as Chaikin's subdivision rule for curves extended to surfaces, we first explain geometrically how to determine the Chaikin local reverse subdivision rule.

Chaikin's subdivision is a "corner-cutting" strategy first presented in [5]. Figure 1 shows one step of this subdivision as a two-step process (for simplicity, we use v_i to stand for ${v_i}^k$ and w_j for ${v_j}^{k+1}$). Two new points are first determined on every line segment, and then the second new point on each successive line segment is connected with the first new point of the next segment. It can be shown that the points generated by repeatedly applying Chaikin's subdivision rule

converge to a quadratic B-spline curve, and thus the limiting curve is $\mathcal{C}^1$ (smooth) [15].

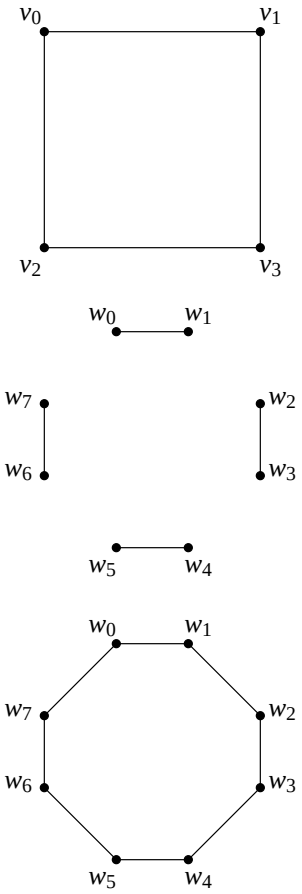


Figure 1: *A two stage representation of Chaikin's rule.*

3.1. Chaikin's Local Reverse Subdivision

In a subdivision process, the v_i are assumed to be given and the w_j are computed. In a reverse subdivision process, however, to obtain lower resolutions from the higher ones, it is assumed that the w_j are given and the v_i are needed. A possible situation is depicted in Figure 2, in which the formula relating w_{2i-2}, w_{2i-1}, and v_i is solved for v_i, as is the formula relating w_{2i}, w_{2i+1}, and v_i.

From this, two candidates, $\overset{\ell}{v}_i$ and $\overset{r}{v}_i$, suggest themselves for replacing each v_i along the curve. It is clear that if all w_j local to v_i come through a subdivision scheme, then we would want to have $\overset{\ell}{v}_i = \overset{r}{v}_i = v_i$. But, in general and computationally, we can not expect that these values coincide. However, it can be shown that if we let v_i be the average of these two candidates, $v_i = \frac{1}{2}\left(\overset{\ell}{v}_i + \overset{r}{v}_i\right)$, then v_i will approximate both candidates locally in the least squares sense. The proof follows the outline we give in Sections 5 and 6, respectively for 4-sided and 3-sided surface meshes.

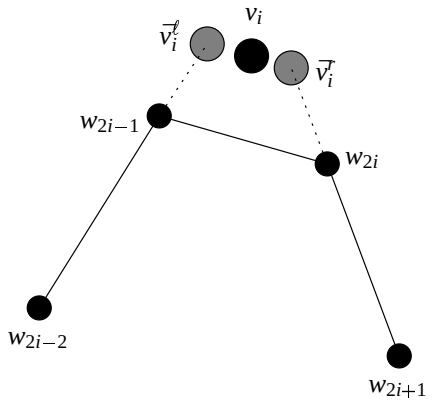


Figure 2: *A situation for Chaikin's local reverse subdivision.*

4. Doo Subdivision

Doo subdivision is a "corner-cutting" method , presented in [6], for the representation of three dimensional surfaces. In this method, the shapes of the faces of the surface are assumed to be arbitrary (consequently, if a face is not triangular, it may not be planar). It can be shown that the Doo subdivision scheme is an extension of Chaikin's method. To show this, consider Chaikin's method in two stages as shown in Figure 1.

In the first stage, contract each line segment of the polygon towards its *center* by a factor α. For example, in Figure 1, (v_0, v_1) is contracted into (w_0, w_1), where $w_0 = \alpha d + (1-\alpha)v_0$, $d = \frac{v_0+v_1}{2}$, and so on for the other w points. In the second stage, for those line segments having been joined together before the subdivision, introduce a new line segment by joining the corresponding end points of their contracted line segments. Again in Figure 1, the line segment (w_0, w_1) is joined with (w_2, w_3) through the new line segment (w_1, w_2). The multiplying factor α should be so chosen that the limiting curve be $\mathcal{C}^1$. For instance $\alpha = \frac{1}{2}$ leads to a $\mathcal{C}^1$ curve.

For the Doo subdivision of surfaces, the operations above on the line segments of a polygon are extended to the faces of a polyhedron: the faces are contracted towards their *centroids*, and the contracted versions of adjoining faces are joined by introducing additional new faces. For example, in Figure 3 the original surface is composed of the two adjoining faces f_1 and f_2. In the first stage, g_1 and g_2 are obtained by the contraction of f_1 and f_2. In the next stage, g_1 and g_2 are joined through g_3. To observe the effect on a three dimensional shape, see Figure 21, where three Doo subdivisions are performed on a cube.

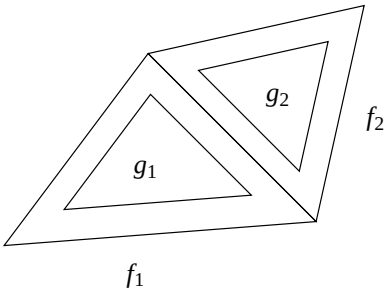


Figure 3: *A simple Doo subdivision.*

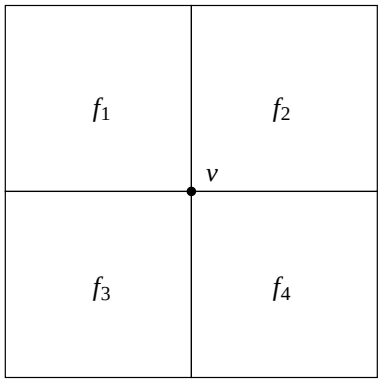
To explain more completely, assume that (F^k, V^k) is a surface of an arbitrary topological type, being represented by faces F^k and vertices V^k. Each vertex $v \in V^k$ is identified by its coordinates in space and each face by the sequence of its associated vertices. Let E be the set of the existing edges in (F^k, V^k). We explain how to identify a subdivided surface (F^{k+1}, V^{k+1}) as the result of an application of the Doo subdivision scheme on (F^k, V^k). First, for each $f_i \in F^k$ introduce a new face $g_i \in F^{k+1}$ by contraction, as in Figure 3, in the following way: If $f_i = (v_1, \ldots, v_n)$ is the face in original faces, and $g_i = (w_1, \ldots, w_n)$, is the contracted face, the w_μ come from contracting the v_λ with respect to the centroid d of f_i as follows (for any reasonable, systematic indexing system for the f, g, v, and w):

$$\begin{cases} w_\mu = (1-\alpha)d + \alpha v_\lambda \\ d = \frac{1}{n}(v_1 + v_2 + \ldots + v_n). \end{cases} \tag{5}$$

In (5), the value of α, $0 < \alpha < 1$, can be chosen so that the limiting surface will be $\mathcal{C}^1$ (see [6, 13], for example, the value $\alpha = \frac{1}{2}$ implies this property). After this operation, the set of all w_μ obtained from the contraction of all faces, forms the new set V^{k+1}. Note that each $v \in V^k$ corresponds to $deg(v)$ (the *degree* of v; that is, the number of faces joined at v) new vertices introduced in V^{k+1} (see Figure 4).

The set of new faces in F^{k+1} is composed of the following three types.

- Face-to-face: The set of new faces, g_i, obtained from F^k, denoted by F_F.
- Face-to-edge: If f_i is joined with f_j through an edge e in the graph (F^k, V^k), then we introduce a face g_e joining g_i and g_j in (F^{k+1}, V^{k+1}) (as g_3 was introduced in Figure 5, for example) so that g_i is joined with g_e and g_e is joined with g_j. Thus, for each edge in (F^k, V^k), not lying on the boundary, a new face is introduced in F^{k+1}. We denote these faces by F_E.
- Face-to-vertex: For each internal vertex v in (F^k, V^k), a new face g_v is introduced so that it joins the corresponding new vertices w_j of v (as in Figure 5). We collect these faces in F_V.



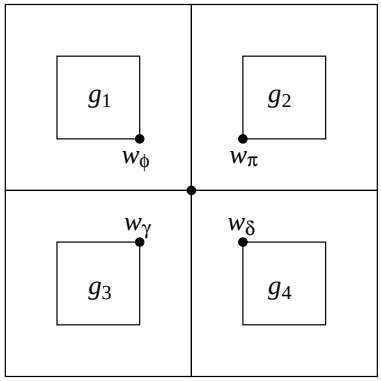


Figure 4: *Conversion of v to w_ϕ, w_π, w_γ, and w_δ.*

Notes: It can easily be verified that

- each face in F_F has the same number of edges as the face from which it is contracted,
- each face in F_E is rectangular (has four sides), and
- the number of sides for each face in F_V is equal to the degree of its corresponding vertex in V^k.

Determining these three types of faces, the subdivision is complete. Therefore, we have

$$F^{k+1} = F_F \cup F_E \cup F_V.$$

5. Quadrilateral Faces

We show in Sections 5 and 6 an average-based reversing process for two special cases of regular face structures. The result of the average reversing will be found to be the same as the result of the local, least squares reversing process in [1] for those face structures. In Section 7, we extend the average-based reversing for any arrangement of faces in Doo subdivision.

We establish in this section that, for a geometry of adjacent quadrilateral faces, reversing Doo subdivision by means

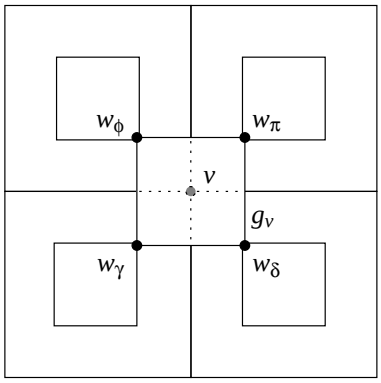


Figure 5: *g_v, the face associated with v.*

of local averaging is equivalent to reversing the subdivision by a local least squares estimate [1] . In Section 6 we establish the same thing for a geometry of adjacent triangular faces.

Consider a regular portion of a rectangular mesh for Doo Subdivision as in Figure 6.

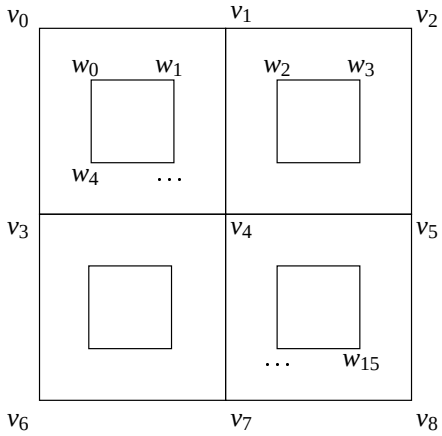


Figure 6: *A regular, rectangular setting.*

We have numbered the coarse vertices v_i from left to right and top to bottom. The fine vertices w_j come from Doo Subdivision, and they have been numbered in the same way. We can express Doo Subdivision locally by a matrix relation $W = PV$ with

$$W = [w_0, w_1, \ldots, w_{15}]^T$$

$$V = [v_0, v_1, \ldots, v_8]^T$$

5.1. Elements of P

Consider Figure 7. We have:

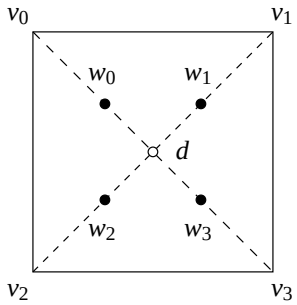


Figure 7: *Contraction on one rectangle.*

$$d = \frac{1}{4}(v_0 + v_1 + v_3 + v_4)$$

and, using $\alpha = \frac{1}{2}$ (for simplicity – other values will produce the similar results):

$$\begin{aligned} w_0 &= (1-\alpha)v_0 + \alpha d = \frac{1}{2}v_0 + \frac{1}{2}d \\ &= \frac{5}{8}v_0 + \frac{1}{8}v_1 + \frac{1}{8}v_3 + \frac{1}{8}v_4 \end{aligned}$$

It can easily be verified that:

$$w_1 = \frac{1}{8}v_0 + \frac{5}{8}v_1 + \frac{1}{8}v_3 + \frac{1}{8}v_4$$

and shifted versions of this relation hold for the other w. Thus, the numbers $\frac{5}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}$ repeat in each row of P and their positions relate to indexing of coarse and fine vertices. That is, P is given by

$$\begin{bmatrix}
5/8 & 1/8 & 0 & 1/8 & 1/8 & 0 & 0 & 0 & 0 \\
1/8 & 5/8 & 0 & 1/8 & 1/8 & 0 & 0 & 0 & 0 \\
0 & 5/8 & 1/8 & 0 & 1/8 & 1/8 & 0 & 0 & 0 \\
0 & 1/8 & 5/8 & 0 & 1/8 & 1/8 & 0 & 0 & 0 \\
1/8 & 1/8 & 0 & 5/8 & 1/8 & 0 & 0 & 0 & 0 \\
1/8 & 1/8 & 0 & 1/8 & 5/8 & 0 & 0 & 0 & 0 \\
0 & 1/8 & 1/8 & 0 & 5/8 & 1/8 & 0 & 0 & 0 \\
0 & 1/8 & 1/8 & 0 & 1/8 & 5/8 & 0 & 0 & 0 \\
0 & 0 & 0 & 5/8 & 1/8 & 0 & 1/8 & 1/8 & 0 \\
0 & 0 & 0 & 1/8 & 5/8 & 0 & 1/8 & 1/8 & 0 \\
0 & 0 & 0 & 0 & 5/8 & 1/8 & 0 & 1/8 & 1/8 \\
0 & 0 & 0 & 0 & 1/8 & 5/8 & 0 & 1/8 & 1/8 \\
0 & 0 & 0 & 1/8 & 1/8 & 0 & 5/8 & 1/8 & 0 \\
0 & 0 & 0 & 1/8 & 1/8 & 0 & 1/8 & 5/8 & 0 \\
0 & 0 & 0 & 0 & 1/8 & 1/8 & 0 & 5/8 & 1/8 \\
0 & 0 & 0 & 0 & 1/8 & 1/8 & 0 & 1/8 & 5/8
\end{bmatrix}$$

5.2. Local Least Squares Reversing

To employ our method in [1] we must determine v_4 from solving the local least squares problem given by $PV = W$. The subdivision matrix P usually is a rectangular matrix, and consequently, it isn't invertible. The pseudoinverse of P, defined by $(P^TP)^{-1}P^T$ and denoted by P^+, is a generalization of inverse matrix when P is not invertible [9]. We have $V = P^+W$, and the vector v_4 would correspond to extracting the fifth row of $V = P^+W$. Using a symbolic algebra system to find the pseudoinverse of the matrix P just given, we find that this produces

$$\begin{aligned} v_4 \;=\; & (-\tfrac{1}{16})w_0+(-\tfrac{1}{16})w_1+(-\tfrac{1}{16})w_2+(-\tfrac{1}{16})w_3+\\ & (-\tfrac{1}{16})w_4+(\tfrac{7}{16})w_5+(\tfrac{7}{16})w_6+(-\tfrac{1}{16})w_7+\\ & (-\tfrac{1}{16})w_8+(\tfrac{7}{16})w_9+(\tfrac{7}{16})w_{10}+(-\tfrac{1}{16})w_{11}+\\ & (-\tfrac{1}{16})w_{12}+(-\tfrac{1}{16})w_{13}+(-\tfrac{1}{16})w_{14}+(-\tfrac{1}{16})w_{15} \end{aligned}$$

We can use a geometric interpretation as in Figure 8. Note that vertices w near v_4 have $\frac{7}{16}$ as coefficient and vertices w far from v_4 have $-\frac{1}{16}$ as coefficient. The sum of all coefficients is unity.

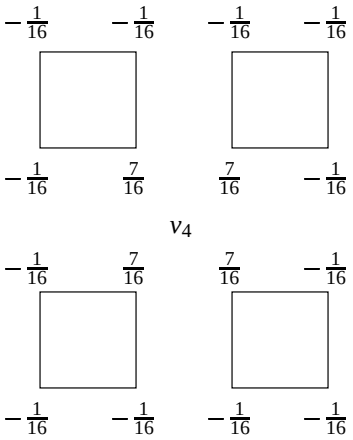


Figure 8: *Diagram of the local averaging.*

5.3. Average Reversing

Consider using an averaging process to reverse subdivision for that same example (Figure 7). We propose the reversal process to be:

- Determine the centroids d_1, d_2, d_3, d_4 of faces g_1, g_2, g_3, g_4 as in Figure 9.
- Expand the faces g_1, g_2, g_3, g_4 with respect to their centroids.
- Determine all the candidates of v_4 (call them $\bar{v}_5$, $\bar{v}_6$, $\bar{v}_9$, $\bar{v}_{10}$ corresponding to their associated w vertices).

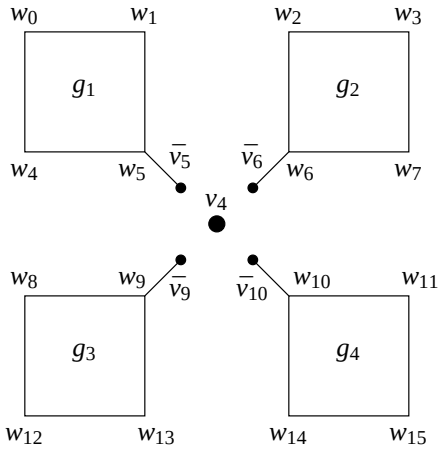


Figure 9: *The averaging process for rectangular faces.*

- Let v_4 be the average of $\bar{v}_5$, $\bar{v}_6$, $\bar{v}_9$, $\bar{v}_{10}$.

Following this plan, we have:

- Centroids

$$\begin{aligned} d_1 &= \frac{1}{4}(w_0+w_1+w_4+w_5)\\ d_2 &= \frac{1}{4}(w_2+w_3+w_6+w_7)\\ d_3 &= \frac{1}{4}(w_8+w_9+w_{12}+w_{13})\\ d_4 &= \frac{1}{4}(w_{10}+w_{11}+w_{14}+w_{15}) \end{aligned}$$

- Expansion

$$\begin{aligned} \bar{v}_5 &= 2w_5-d_1=\frac{7}{4}w_5-\frac{1}{4}(w_0+w_1+w_4)\\ \bar{v}_6 &= 2w_6-d_2=\frac{7}{4}w_6-\frac{1}{4}(w_2+w_3+w_7)\\ \bar{v}_9 &= 2w_9-d_3=\frac{7}{4}w_9-\frac{1}{4}(w_8+w_{12}+w_{13})\\ \bar{v}_{10} &= 2w_{10}-d_4=\frac{7}{4}w_{10}-\frac{1}{4}(w_{11}+w_{14}+w_{15}) \end{aligned}$$

- Average

$$\begin{aligned} v_4 &= \frac{1}{4}(\bar{v}_5+\bar{v}_6+\bar{v}_9+\bar{v}_{10})\\ &= \frac{7}{16}(w_5+w_6+w_9+w_{10})\\ &\quad -\frac{1}{16}(w_0+w_1+w_4+w_2+w_3+w_7\\ &\qquad +w_8+w_{12}+w_{13}+w_{11}+w_{14}+w_{15}) \end{aligned}$$

The result is exactly the same as from the local least squares approach.

6. Triangular Faces

Consider a regular portion of a triangular mesh for Doo Subdivision as in Figure 10.

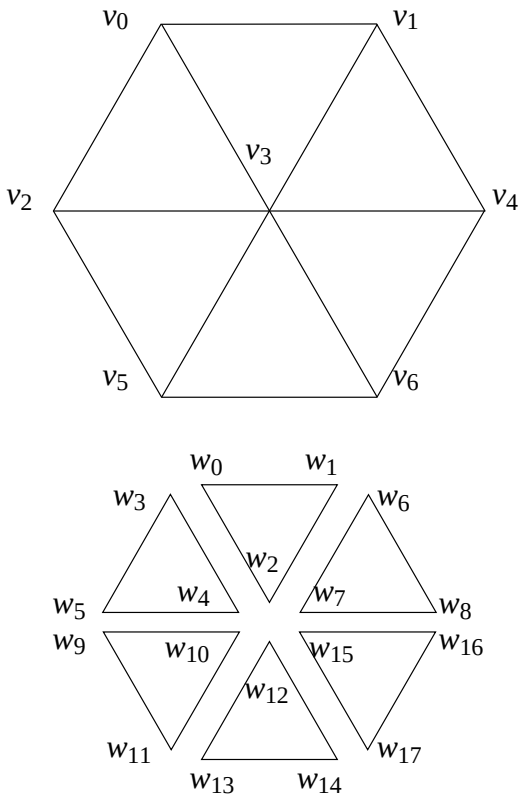


Figure 10: *Coarse and fine vertices for triangular faces.*

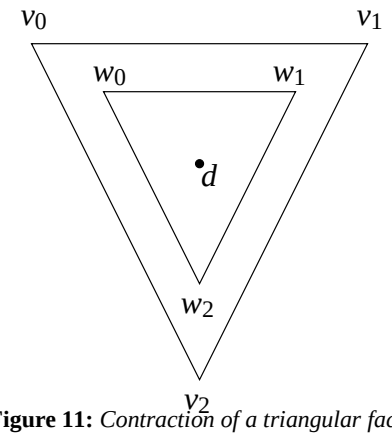


Figure 11: *Contraction of a triangular face.*

6.1. Elements of P

Consider Figure 11, if we use $\alpha = \frac{1}{2}$ again, we will get

$$\begin{aligned} d &= \frac{1}{3}(v_0 + v_1 + v_2) \\ w_0 &= \frac{1}{2}v_0 + \frac{1}{2}d = \frac{4}{6}v_0 + \frac{1}{6}v_1 + \frac{1}{6}v_2 \end{aligned}$$

It can easily be verified that $\frac{4}{6}, \frac{1}{6}, \frac{1}{6}$ appear in every relation, so they occur in every row of P.

6.2. Local Least Squares Reversing

P has the form:

$$\begin{bmatrix} 2/3 & 1/6 & 0 & 1/6 & 0 & 0 & 0 \\ 1/6 & 2/3 & 0 & 1/6 & 0 & 0 & 0 \\ 1/6 & 1/6 & 0 & 2/3 & 0 & 0 & 0 \\ 2/3 & 0 & 1/6 & 1/6 & 0 & 0 & 0 \\ 1/6 & 0 & 2/3 & 1/6 & 0 & 0 & 0 \\ 1/6 & 0 & 1/6 & 2/3 & 0 & 0 & 0 \\ 0 & 2/3 & 0 & 1/6 & 1/6 & 0 & 0 \\ 0 & 1/6 & 0 & 2/3 & 1/6 & 0 & 0 \\ 0 & 1/6 & 0 & 1/6 & 2/3 & 0 & 0 \\ 0 & 0 & 2/3 & 1/6 & 0 & 1/6 & 0 \\ 0 & 0 & 1/6 & 2/3 & 0 & 1/6 & 0 \\ 0 & 0 & 1/6 & 1/6 & 0 & 2/3 & 0 \\ 0 & 0 & 0 & 2/3 & 0 & 1/6 & 1/6 \\ 0 & 0 & 0 & 1/6 & 0 & 2/3 & 1/6 \\ 0 & 0 & 0 & 1/6 & 0 & 1/6 & 2/3 \\ 0 & 0 & 0 & 2/3 & 1/6 & 0 & 1/6 \\ 0 & 0 & 0 & 1/6 & 2/3 & 0 & 1/6 \\ 0 & 0 & 0 & 1/6 & 1/6 & 0 & 2/3 \end{bmatrix}$$

Setting up the least squares system and extracting the row corresponding to v_3 yields:

$$\begin{aligned} v_3 &= \frac{5}{18}(w_2 + w_5 + w_7 + w_{10} + w_{12} + w_{15}) \\ &\quad - \frac{1}{18}(w_0 + w_1 + w_3 + w_4 + w_6 + w_8 \\ &\qquad + w_9 + w_{11} + w_{15} + w_{14} + w_{16} + w_{17}) \\ v_3 &= \frac{5}{18}(nearvertices) - \frac{1}{18}(farvertices) \end{aligned}$$

6.3. Average Reversing

If instead we propose a reversal based upon averaging to determine v_3, the prescription would be:

- Centroids

$$\begin{aligned} d_1 &= \frac{1}{3}(w_0 + w_1 + w_2) \\ d_2 &= \frac{1}{3}(w_3 + w_4 + w_5) \end{aligned}$$

$$
\begin{aligned}
d_3 &= \frac{1}{3}(w_6 + w_7 + w_8) \qquad (6)\\
d_4 &= \frac{1}{3}(w_9 + w_{10} + w_{11})\\
d_5 &= \frac{1}{3}(w_{12} + w_{13} + w_{14})\\
d_6 &= \frac{1}{3}(w_{15} + w_{16} + w_{17})
\end{aligned}
$$

- Expansion

$$
\begin{array}{llll}
\bar{\nu} = w_2 - d_1 & \bar{\nu}_0 = 2w_{10} - d_4\\
\bar{\nu} = 2w_5 - d_2 & \bar{\nu}_2 = 2w_{12} - d_5\\
\bar{\nu} = 2w_7 - d_3 & \bar{\nu}_5 = 2w_{15} - d_6
\end{array}
$$

If we replace d_i by (6) we will get

$$
\begin{aligned}
\bar{\nu} &= 2w_2 - \frac{1}{3}(w_0 + w_1 + w_2) = \frac{5}{3}w_2 - \frac{1}{3}w_0 - \frac{1}{3}w_1\\
\bar{\nu} &= \frac{5}{3}w_5 - \frac{1}{3}w_3 - \frac{1}{3}w_4\\
\bar{\nu} &= \frac{5}{3}w_7 - \frac{1}{3}w_6 - \frac{1}{3}w_8\\
\bar{\nu}_0 &= \frac{5}{3}w_{10} - \frac{1}{3}w_9 - \frac{1}{3}w_{11}\\
\bar{\nu}_2 &= \frac{5}{3}w_{12} - \frac{1}{3}w_{13} - \frac{1}{3}w_{14}\\
\bar{\nu}_5 &= \frac{5}{3}w_{15} - \frac{1}{3}w_{16} - \frac{1}{3}w_{17}
\end{aligned}
$$

- Average

$$
\begin{aligned}
v_3 &= \frac{1}{6}(\bar{\nu} + \bar{\nu} + \bar{\nu} + \bar{\nu}_0 + \bar{\nu}_2 + \bar{\nu}_5)\\
v_3 &= \frac{5}{18}(w_2 + w_5 + w_7 + w_{10} + w_{12} + w_{13})\\
&\quad - \frac{1}{18}(w_0 + w_1 + w_3 + w_4 + w_6 + w_8\\
&\qquad + w_9 + w_{11} + w_{13} + w_{14} + w_{16} + w_{17})
\end{aligned}
$$

This determines v_3 as exactly the same as the result of local least squares reversal.

7. Reverse Doo Subdivision for General Face Geometry

We develop the multiresolution of surfaces having arbitrary topologies by presenting an efficient method for reversing the Doo subdivision. We saw how to construct a finer resolution (F^{k+1}, V^{k+1}) by an application of the Doo subdivision to (F^k, V). F^{k+1} is composed of three types of new faces: face-to-face, face-to-edge and face-to-vertex. To reverse the process, we assume that (F^{k+1}, V^{k+1}) is at hand and consider constructing an appropriate (F^k, V). We know that each face in F_F is supposed to be obtained as a result of the contraction of a corresponding face in F^k. Therefore, with F_F at hand, we can expand the faces, with respect to their centroids, to their originals in F^k. But, in general, it is not mandatory that (F^{k+1}, V^{k+1}) be obtained directly as a result of a subdivision. To be more precise, we may assume that the coordinates of the vertices in V^{k+1}, for some reasons (for instance, computationally), are not exactly those arrived at by a subdivision. So we expect that the faces obtained by the expansion may not join exactly. For an illustration, see Figuress 12 and 13. These figures suggest the approach to be taken. A subset of the F^{k+1} are to be identified as F_F faces, and each member of the subset is to be expanded with respect to its centroid in order to provide the set F^k of faces for the coarse surface.

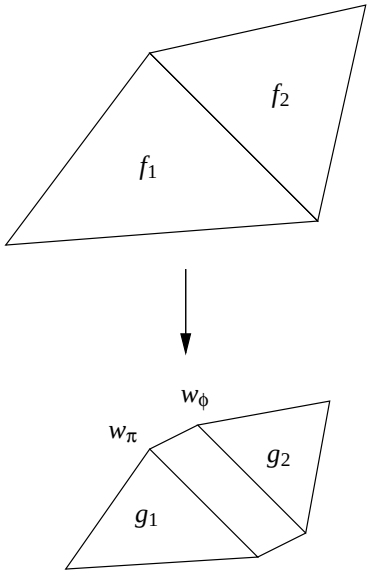


Figure 12: *Doo's subdivision in a simple figure.*

Figure 12 shows a simple surface along with its corresponding subdivision in which g_1 and g_2 are the contraction of f_1 and f_2 respectively, vertices w_π and w_ϕ each correspond to v. In Figure 13, the up figure is assumed to be of a higher resolution, which may possess inaccuracies, while the down figure is the situation being encountered after an application of a reverse subdivision. We see that two candidates $\bar{\nu}$ and $\bar{\nu}$, are introduced to be placed for v. This situation is similar to the one depicted in Figure 2, the result of a reverse Chaikin's subdivision. Thus, in general, for each $v \in V$, we may expect several candidates (i.e., the number of faces to be joined at v) to arise after an expansion of the corresponding faces in F_F. As in our reverse Chaikin's subdivision method, and as suggested by the discussions in Sections 5 and 6, we let the vertex be set to the average of its candidates.

Considering the above, we outline the following stages for carrying out a reverse Doo subdivision process:

- Identify faces F_F in F^{k+1}.
- Expand the faces in F_F.
- Determine all the candidates for each potential vertex in V^k.
- Determine the coordinates of all vertices in V^k.

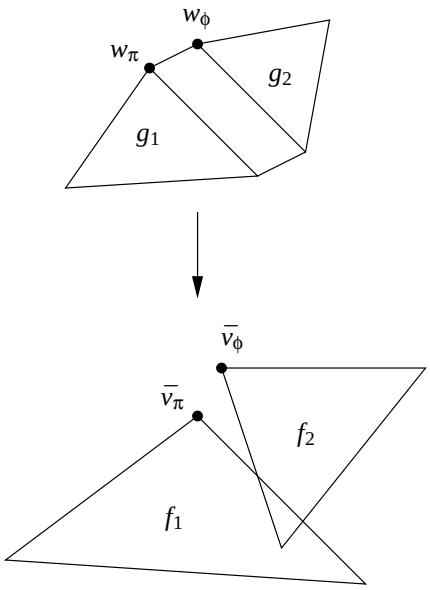


Figure 13: *Disconnection of faces resulting from expansion of F_F.*

We emphasize that (F^{k+1}, V^{k+1}) must have the subdivision connectivity property, which means (F^{k+1}, V^{k+1}) must qualify as a graph obtainable by a subdivision (see Figure 14) . We note that the exact subdivision may follow the subdivision path as in the Figure, but the graph (F^{k+1}, V^{k+1}) at hand must be at least an edited (or computed) version of the actual subdivided graph $(\overline{F}^{k+1}, \overline{V}^{k+1})$ so that the reverse subdivision path may be followed.

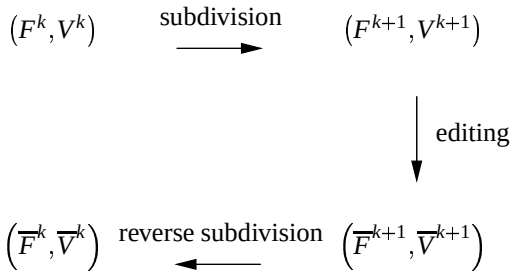


Figure 14: *The change in coordinates of the vertices through subdivisions.*

Therefore, although there are no restrictions on the values of the coordinates of the vertices in V^{k+1}; nevertheless, the faces in F^{k+1} are required to be arranged (in regard to adjacencies) so that they can be generated by a Doo subdivision of a hypothetical surface. This requirement of subdivision connectivity can be justified by most practical applications.

7.1. Identifying Faces F_F in F^{k+1}

We know that

$$F^{k+1} = F_F \cup F_V \cup F_E.$$

We also know that, in the subdivision process for converting F^k to F^{k+1}, F_F is obtained at the first step and in the next steps, F_E and F_V will be obtained consequently. Therefore, we can put these types of faces into F^{k+1} with the same order that are obtained, permitting us to write:

$$F^{k+1} = (F_F, F_E, F_V)$$

As the result of this, the first face of F^{k+1} will certainly be in F_F. Therefore, we could use this fact for the reversal process in converting F^{k+1} to F^k. We could additionally use any special information we might have regarding F_F and F_E in F^{k+1}. But, to be more efficient and more general, we present a method which requires the knowledge of only one face in F_F and finds the rest by a linear search.

Let $g \in F_F$. We wish to identify the remaining elements of F_F from the ones in F^{k+1}. We notice that if there exists a rectangular face $\tilde{g}_i \in F_E$ between g and some face g_i, then g_i belongs to F_F. We use Figure 15, as an example, to make our points. We observe that the face $\tilde{g}_1$ is located between faces g and g_1, meaning that $\tilde{g}_1$ has a common edge with g and another common edge with g_1, but g and g_1 have no common edge or vertex. Thus, g_1 belongs to F_F as well (in fact, g and g_1 were two adjacent faces of the surface before the subdivision, and the face $\tilde{g}_1$ is produced after the subdivision due to the existence of a common edge between g and g_1). The same observation applies to g_2, g_3, and g_4. Therefore, $\tilde{g}_1$, $\tilde{g}_2$, $\tilde{g}_3$, and $\tilde{g}_4$ are faces of the face-to-edge type belonging to F_E.

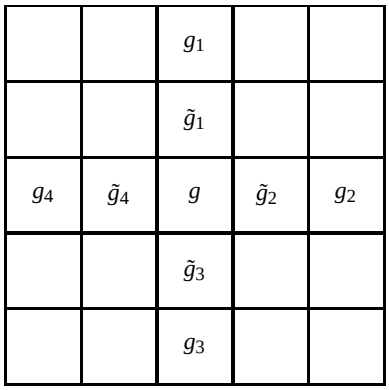


Figure 15: *The faces of F^{k+1}.*

With these considerations, we interpret the problem of finding the elements of F_F as a graph traversal. We first define graph G_F as follows. The vertices of G_F correspond to the faces F_F (with the first face to start with, as the root), and the edges correspond to the faces in F_E. For example, Figure 16 shows the corresponding graph G_F for Figure 15

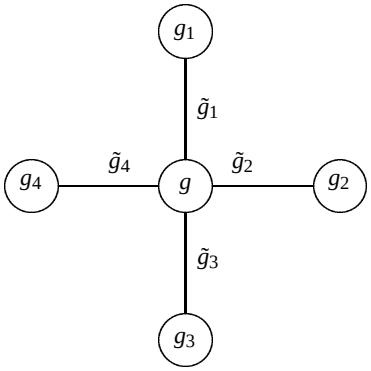


Figure 16: *The graph associated with F^{k+1}.*

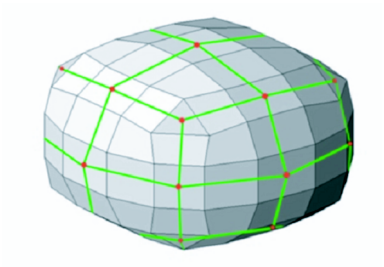


Figure 17: *The graph associated with F^{k+1} A surface example and its G_F graph .*

and in Figure 17 shows a surface and its G_F Graph . We can see that the cost for finding the faces F_F corresponds to the cost for performing a breadth (or a depth) first search of the vertices of G_F.

The main operations for the traversal is of $O(n_e)$, where n_e is the number of edges in G_F. Considering that the number of edges in G_F equals the number of faces in F_E, and knowing that

$$|F_E| \leq |F^{k+1}| \leq |V^{k+1}|,$$

we realize $n_e \leq |V^{k+1}|$. We conclude that the proposed traversal cost is only linearly dependent on the number of vertices of the existing surface. Therefore, the cost for finding F_F is linear in the data.

7.2. Expansion of the Faces F_F

To construct F^k, we need to expand the faces in F_F with respect to their centroids (reversing the contraction process (5)). We use the following notations:

- g for a face in F_F having some m sides.
- $w_1, \ldots, w_m$ for the vertices of the face g.
- d for the centroid of g.
- f for the face obtained after the expansion of g.
- $\bar{v}_1, \ldots, \bar{v}_m$ for vertices of the face f.

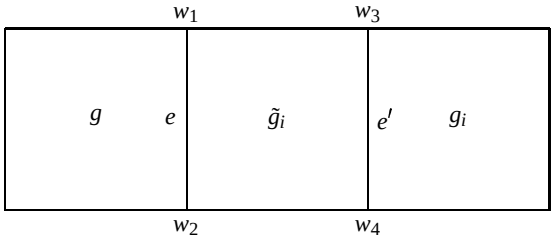


Figure 18: *Adjacency of g, $\tilde{g}_i$ and g_i.*

- α for the contraction factor in Doo subdivision (for example $\alpha = \frac{1}{2}$ is suitable)

We note that the expansion factor for g equals $\frac{1}{\alpha}$. With these notations, we observe:

- The centroids of f and g coincide, i.e., d is the centroid of both.
- The following relations hold for the vertices of f and g:

$$\bar{v}_i = \frac{1}{\alpha} w_i - \frac{1-\alpha}{\alpha} d \qquad i = 1, 2, \ldots, m. \tag{7}$$

Therefore, having g and using (7), we can determine the vertices of the face f.

7.3. Determining the Candidates for the Vertices in V^k

Suppose that the faces g and g_i are adjoined by the face $\tilde{g}_i$ (as in Figure 18). Then, there exist some edges $e = (w_1, w_2)$ common to g and $\tilde{g}_i$, and $e' = (w_3, w_4)$ common to g_i and $\tilde{g}_i$. After the expansion of g and g_i, the pair of vertices $\bar{v}_3$ and $\bar{v}_1$, and $\bar{v}_4$ and $\bar{v}_2$ are presumed to coincide. As explained before, this is more likely not to occur. Thus, $\bar{v}_3$ and $\bar{v}_1$ serve as two candidates for one vertex in V^k, and $\bar{v}_4$ and $\bar{v}_2$ for another. We show, by properly examining these situations during the process of finding faces in F_F (or traversing the vertices of G_F), how to identify the various candidates for each vertex in V^k. We construct another graph G_C whose vertices are the same as the ones in V^{k+1}. In every phase of the traversal of G_F, we add two edges to G_C. For instance, if $\bar{v}_1$ and $\bar{v}_3$ are two candidates for a vertex and $\bar{v}_2$ and $\bar{v}_4$ are two candidates for another vertex, then we include the respective edges $(\bar{v}_1, \bar{v}_3)$ and $(\bar{v}_2, \bar{v}_4)$ in G_C. After the completion of the traversal of G_F, the graph G_C is completely defined. Figure 19 shows a surface (F^{k+1}, V^{k+1}) on the up and its corresponding graph G_C on the down. Now, it is observed that the vertices of a connected component in G_C are the candidates to be replaced by a single vertex $v \in V$. The connected components of a graph may be found by a breadth first search also. So, the operational cost is $O(n_e)$, where n_e is the number of edges in G_C [2].

We note that in every phase of the traversal of the nodes of G_F, an association of two faces in F_F are established through a face in F_E (see Figure 18, for an example). Thus, for every

face in F_E, there corresponds two edges in G_C. Hence, n_e, the number of edges in G_C, is twice the number of face-to-edge faces F_E. Therefore, we have

$$n_e = 2|F_E| \leq 2|V^{k+1}|.$$

In other words, the main operational cost for determining the connected components of G_C is linear, with respect to $|V^{k+1}|$.

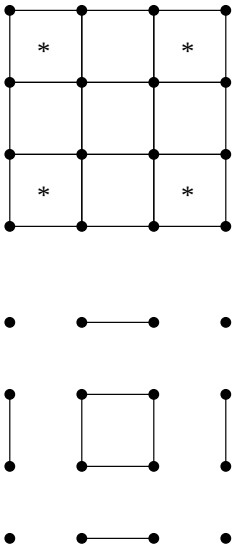


Figure 19: *The surface (F^{k+1}, V^{k+1}) above and the associated graph G_C below.*

7.4. Determining the Vertices V^k

We saw that all the vertices of V^{k+1} belonging to a connected component in G_C must be replaced by a vertex $v \in V$. Suppose $w_1, w_2, \ldots, w_\ell$ are the vertices in a connected component, and $\bar{w}_1, \bar{w}_2, \ldots, \bar{w}_\ell$ are the corresponding vertices after expansion. If the surface (F^{k+1}, V^{k+1}) were obtained from a Doo subdivision, then we would expect that

$$v_1 = v_2 = \ldots = v_\ell.$$

But, in practice, the v_i are not equal (Figure 20 shows a simple situation in which v_1, v_2, v_3, and v_4 are to be replaced by a vertex). If we let v be the average of $v_1, v_2, \ldots, v_\ell$, then the local error in v is minimized in the least squares sense in certain cases, as we have shown. So, we let

$$v = \frac{1}{\ell}(v_1 + v_2 + \ldots + v_\ell). \tag{8}$$

Repeating the above for all the connected components of G_C, we determine all the vertices in V^k. We remark that the setting (8) for all the vertices in V^k does not necessarily imply an optimal global setting (i.e., does not produce a least global error), but nevertheless, is efficiently practical. In any case, there is no known practical approach for an efficient (linear operational cost) in the global setting [14]

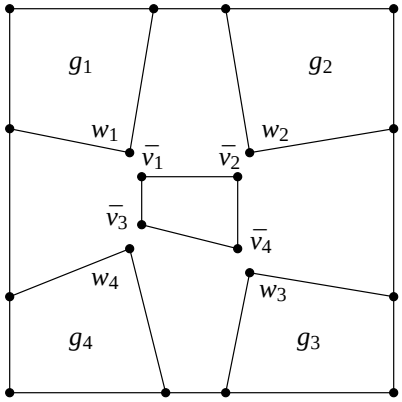


Figure 20: *A simple example of disconnection of the v_i.*

We now present several examples illustrating the effectiveness of our approach. In Figure 21, the result of three successive Doo subdivisions are shown for the cube in the top. In Figure 22, some vertices from the fine surface on the far down of Figure 21 is selected and moved, and three levels of the reverse Doo subdivisions are shown resulting in the original surface, a cube. (The multiresolution implicit in Figure 22 would, of course, include the small peak appearing in the most finely subdivided surface as part of the error representation; that is, in the wavelet portion of the multiresolution.) Figures 23 and 24 show the same processes for another surface.

Figure 27 shows the application of Doo subdivision on the third surface of the Figure 22. Figures 28 and 29 show two other practical examples of the reversing process.

8. Error Representation

We wish to compute and store the local error of (8) so that the reconstruction of (F^{k+1}, V^{k+1}) is made possible. Obviously, if we can reconstruct the points $\bar{w}_i$, then we can easily reconstruct the w_i (equation (7)). We note that the following system of equations is nonsingular:

$$\begin{bmatrix} \mathbf{1} & Q \end{bmatrix} \begin{bmatrix} v \\ \mathbf{e} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{v}} \end{bmatrix} \tag{9}$$

where $\mathbf{1}$ is a column of length ℓ, all of whose entries are 1, where $\bar{\mathbf{v}}$ is the ℓ-length column vector of the $\bar{w}_i$, where $\mathbf{e}$ is a column vector of length $(\ell - 1)$, and, where Q is a matrix of size $\ell \times (\ell - 1)$ whose columns form a basis for the null

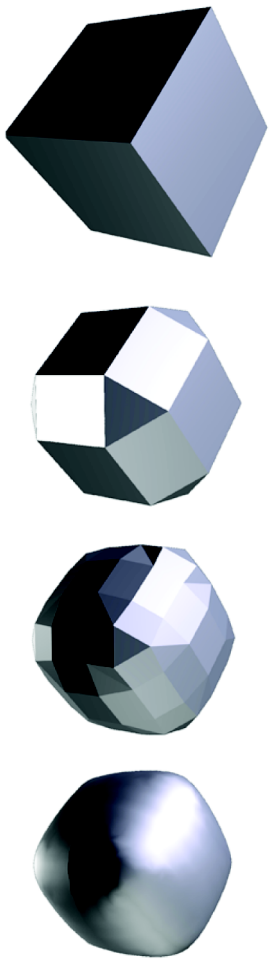


Figure 21: *Three steps of Doo subdivision on a cube.*

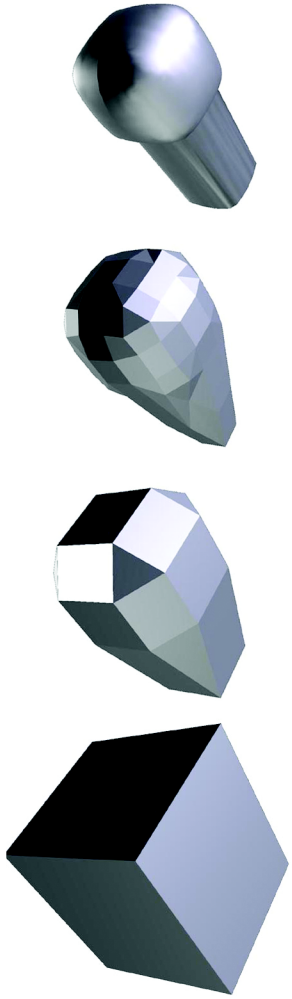


Figure 22: *Reverse Doo subdivision after the change of some vertex coordinates.*

space of **1** – the following choice is quite effective:

$$Q = \begin{bmatrix} 1 & 0 & \dots & 0 \\ -1 & 1 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & -1 \end{bmatrix} \qquad (10)$$

It may be verified that the first row of the inverse of the matrix in equation (10) is $[\frac{1}{\ell}, \dots, \frac{1}{\ell}]$, so the appearance of v as the top element of the "solution vector" of the system is consistent with our definition of it elsewhere as the average of the $\overline{\mathbf{v}}$. By this means, the vertices $(\overline{\mathbf{v}}, \overline{\mathbf{v}}, \dots, \overline{\mathbf{v}})$ are turned into the vertex v along with the error represented by $(e_1, e_2, \dots, e_{\ell-1})$. These error elements may be interpreted as the local wavelet coefficients. We are never concerned with the wavelet functions as such; they are implicitly defined by the matrix Q, just as their corresponding scale functions are defined implicitly by the subdivision P.

We can produce the original fine (F^{k+1}, V^{k+1}) completely. Figures 25 and 26 show the complete reconstruction of the surfaces in Figures 22 and 24 using the error components.

9. Concluding Remarks

We presented an efficient (linear time) reverse Doo subdivision approach, obtaining a multiresolution method for representing surfaces having arbitrary topologies. To decompose a fine surface into a coarse one, we showed how to find the faces of the coarse surface, from among the faces of the fine surface, by a breadth first search of a graph. Then we ex-

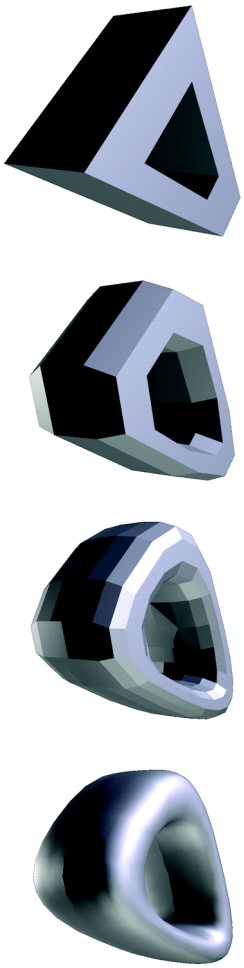


Figure 23: *Doo subdivision.*

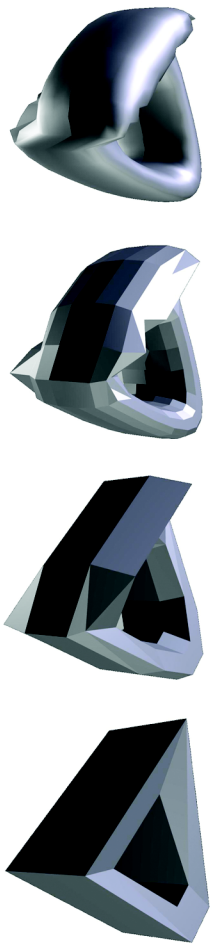


Figure 24: *Reverse Doo subdivision after some changes to vertices.*

panded these faces and obtained candidates for the vertices. Finally, we set each vertex of the coarse surface to the average of its corresponding candidates. We proved that this resulted in a local least squares error, at least for common, regular geometries. We also presented a convenient basis matrix to compute the wavelet coefficients, being utilized for the reconstruction process. We presented several examples illustrating the effectiveness and practicality of our approach.

References

1. R. H. Bartels and F. F. Samavati. Reversing subdivision rules: Local linear conditions and observations on inner products. *Journal of Computational and Applied Mathematics*, 119:29–67, 2000.
2. G. Brassard and G. Bratley. *Fundamentals of Algorithms*. Prentice-Hall, 1996.
3. E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Comput. Aided Des.*, 10(6):350–355, November 1978.
4. Andrew Certain, Jovan Popović, Tony DeRose, Tom Duchamp, David Salesin, and Werner Stuetzle. Interactive multiresolution surface viewing. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 91–98. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.
5. G. Chaikin. An algorithm for high speed curve generation. *Computer Graphics and Image Processing*, 3:346–349, 1974.
6. D. Doo. A subdivision algorithm for smoothing down irregularly shaped polyhedrons. In *Proced. Int'l*

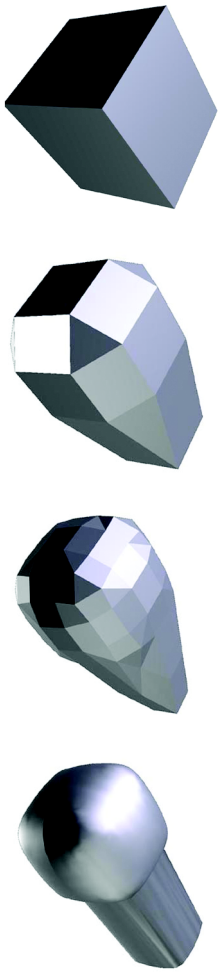


Figure 25: *Complete reconstruction of Figure 22.*

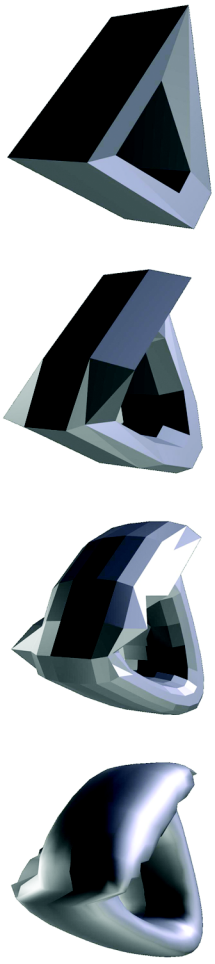


Figure 26: *Complete reconstruction of Figure 24.*

Conf. Ineractive Techniques in Computer Aided Design, pages 157–165, 1978. Bologna, Italy, IEEE Computer Soc.

7. Nira Dyn, David Levin, and John A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, 9(2):160–169, April 1990.

8. M. Garland and P. S. Heckbert. Surface simplification using quadratic error metrics in computer graphics. In *SIGGRAPH '97 Proceedings*, pages 209–218, 1997.

9. G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, second edition, 1989.

10. H. Hoppe. Progressive meshes. In *SIGGRAPH '96 Proceedings*, pages 99–108, August 1996.

11. L. Kobbelt. Interpolatory subdivision on open quadrilateral nets with arbitrary topology. *Computer Graphics Forum*, 15(3):409–420, August 1996. Proceedings of Eurographics '96. ISSN 1067-7055.

12. L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In Michael Cohen, editor, *SIGGRAPH '98 Conference Proceedings*, Annual Conference Series, pages 105–114. ACM SIGGRAPH, Addison Wesley, July 1998.

13. C. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, 1987.

14. M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics*, 16(1):34–73, January 1997.

15. R. Riesenfeld. On chaikin's algorithm. *Computer Graphics and Image Processing*, 4:304–310, 1975.

16. F. F. Samavati and R. H. Bartels. Multiresolution curve and surface representation by reversing subdivision rules. *Computer Graphics Forum*, 18(2):97–119, June 1999.

17. P. Schröder and W. Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 161–172. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.

18. E. J. Stollnitz, T. D. DeRose, and D. H. Salesin. *Wavelets for Computer Graphics*. Morgan Kaufmann Publishers, 1996.

19. D. Zorin, P. Schröder, A. Levin, L. Kobbelt, W. Sweldens, and T. DeRose. Subdivision for modeling and animation. Course Notes, SIGGRAPH 2000.

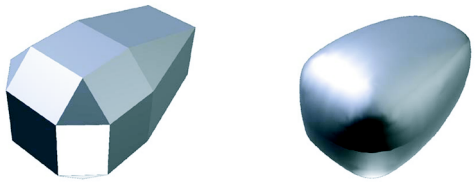


Figure 27: *Applying Doo subdivision, but without adding the error information, to the results of reversal.*

Figure 28: *Dolphin, from left to right: initial coarse model, fine model after applying Doo subdivision, new fine model by editing the Dolphin's tail, and the edited version reversed.*

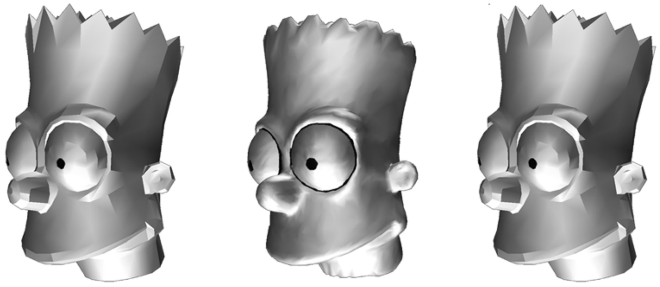


Figure 29: *Bart Simpson, from left to right, initial coarse model, after one level of Doo subdivision, and after one level of reversing. There is no difference between the original and this one.*